An Overview of AP233

STEP's Systems Engineering Standard

October 20, 2003

Jim U'Ren
AP233 Working Group Chair
NASA / JPL

AP233
System Engineering

What is AP233

- **a STEP-based* data exchange standard targeted to support the needs of the systems engineering community,**
- **Parallels emerging standards in MCAD, ECAD, engineering analysis, PDM and product development domains.**
- **Provides neutral data models to exchange and integrate information between systems engineering tools**

* STEP – ISO 10303 (**ST**andard for the **E**xchange of **P**roduct inform

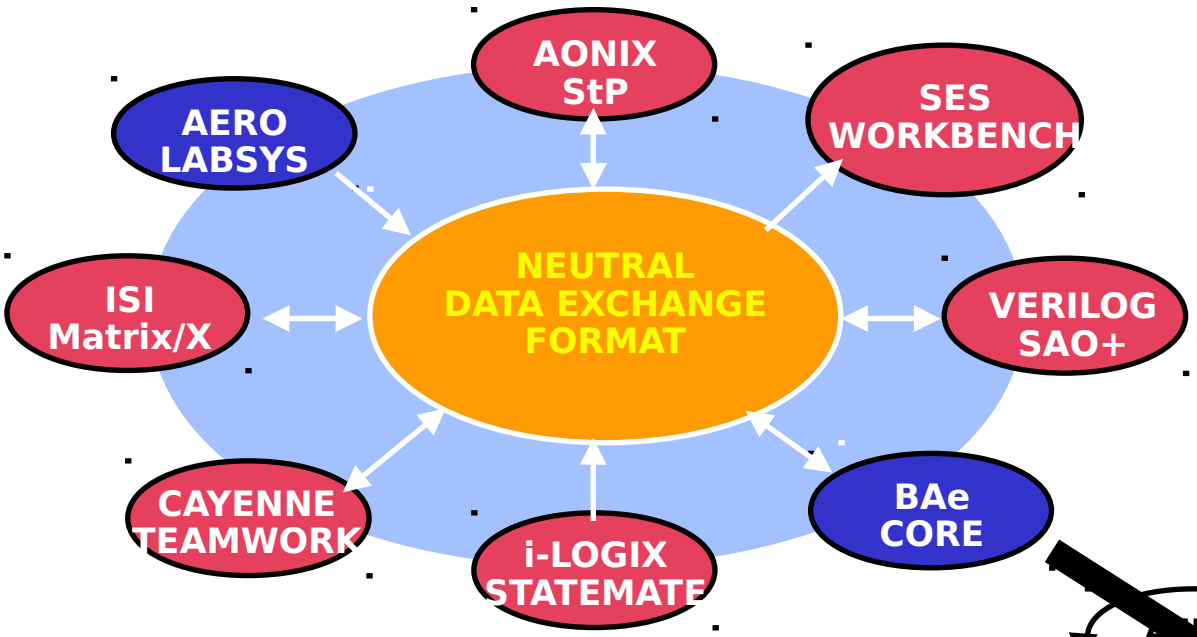
The SEDRES Project:
The Roots of AP233

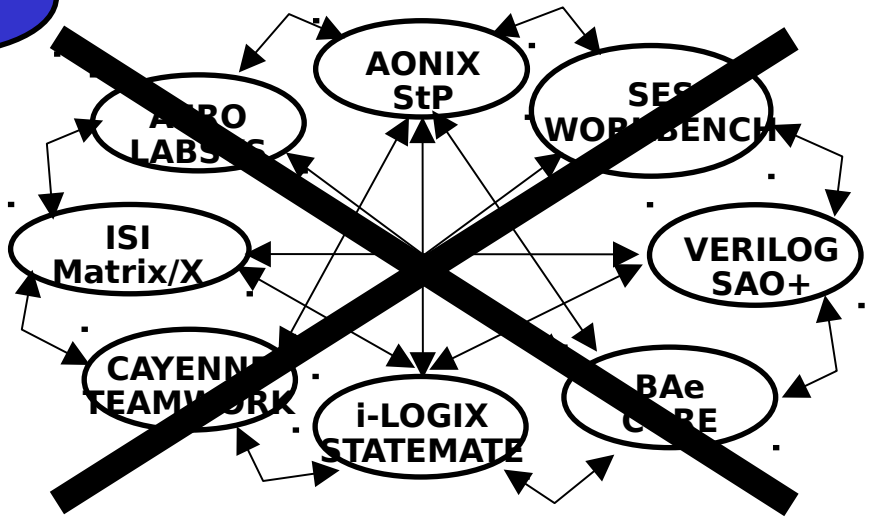
(Systems Engineering Representation and Exchange Standardization)

- **SEDRES Project consisted of European aerospace companies**
  - **Aerospatiale, Alenia, British Aerospace (now BAE Systems), DASA (now DaimlerChrysler), SAAB**
- **Joint projects**
  - **Gripen**
  - **Eurofighter (EF2000)**
- **Project focused on specific SE data exchanges**
- **SEDRES initiated NWI in TC184/SC4 in 1998 to provide a means of publishing its work**
- **Demonstrated that STEP could be used to exchange systems engineering information**

AP233
System Engineering

Primary Driver: Reduction in Tool Interfaces





AP233
System Engineering

Current Approach of AP233 Work

- **Modularization**
  - **Avoids lengthy gestation of large AP efforts**
  - **Allows frequent, sequential deliveries**
  - **Provides on going evolutionary development environment**
- **Active Development Team**
  - **Bi-weekly Telecons**
  - **Web-based repository to collect, secure and publish work-in-progress**
  - **Public website**
- **Collaboration with existing Industry groups**
  - **INCOSE - International Council on Systems Engineering**
  - **OMG - Object Management Group / SE Working Group**

STEP Systems Engineering Team (AP233)

The STEP Systems Engineering Project is coordinated through the PDES Inc., a STEP consortium. Standards organizations collaborating with the Project are INCOSE (International Council for Systems Engineering) and OMG (Object Management Group).

2003-10-20

Planned AP233 Module Sets

- Requirements
- Structural Models
- Behavioral Models
- Validation and Verification
- Data Representation
- Risk Analysis
- Analysis Interfaces
- Scheduling
- Cost Models
- Organizational Structure
- PDM
- Security
- Rules

This slide provides high level information about how STEP and other standards can be applied to the engineering domains that are part of the spacecraft development process.

STEP in Spacecraft Development

How the family of STEP Data Standards can be applied to Spacecraft Development

Fluid Dynamics
- **Standard: AP 237**
- **Status**: In Development
- **Software -** TBD
- **Orgs:** Boeing, AiAA, NASA/Ames)

Propulsion
- **Standard**: STEP-PRP
- **Status**: In Development
- **Software**: TBD
- **Orgs:** ESA, EADS

ElectroMechanical Assembly
- **Standard**: AP210
- **Status**: Prototyped
- **Software:** Mentor Graphics, Eagle, Theorem Solutions, LKSoft, Zuken
- **Orgs:** Rockwell, Boeing, NASA, GaTech, CPES, ATI

Cabling & Wire Harness
- **Standard**: AP212
- **Status**: Prototyped / In production
- **Software** MentorGraphics
- **Orgs:** Daimler-Chrysler, GM, Ford, ABB, ProSTEP, Bosch, Rockwell, Siemens

Optics
- **Standard**: NODIF
- **Status**: In Development
- **Software - TBD**
- **Orgs:** Minolta, Olympus, ORA

Mechanical Engineering
- **Standard**: AP203 Ed. 1 & 2, AP214
- Status: In Production
- **Software** Pro-E, Cadds, SolidWorks, AutoCad, IDEAS, Catia, Unigraphics, Alibre, and others
- **Orgs:** Industry-wide in aerospace and automotive industries (Europe & US)

Software Engineering
- **Standard:**:UML
- **Status**: In Production Industry-wide, a STEP AP233 interface to UML is In Development
- **Software:** Rational Rose, All-Together, Argo, Rhapsody
- **Orgs:** Lockheed, NASA, I-Logix,

Structural Analysis
- **Standard**: AP209
- **Status**: In Production
- **Software**: MSC Patran, Thermal Desktop
- **Orgs:** Lockheed Martin, Airbus, Boeing, NASA Langely

Systems Engineering
- **Standard**: AP233 / STEP-NRF
- **Status:** In development / Prototyped
- **Software**: Core, Doors, MatLab, MS-Excel, RTM, Slate, Statemate, System Arch, TeamWork
- **Orgs:** BAE SYSTEMS, EADS, NASA, CNES, Boeing, Lockheed, Raytheon, Vitech

Thermal Radiation Analysis
- **Standard**: STEP-TAS
- **Status**: In Production
- **Software**: Thermal Desktop, Thermica, ESARAD, TRASYS
- **Orgs:** ESA/ESTEC, NASA (JPL &

PDM
- **Standard:** STEP PDM Schema/AP232
- **Status:** In Production
- **Software:** MetaPhase, Windchill, Insync, Matrix, Share-a-Space, STEP Book, STEP-Tools Inc.
- **Orgs:** Lockheed Martin, EADS, BAE SYSTEMS, Raytheon, NASA, Boeing, Eurostep

De-emphasized boxes indicate data models that are IN DEVELOPMENT.

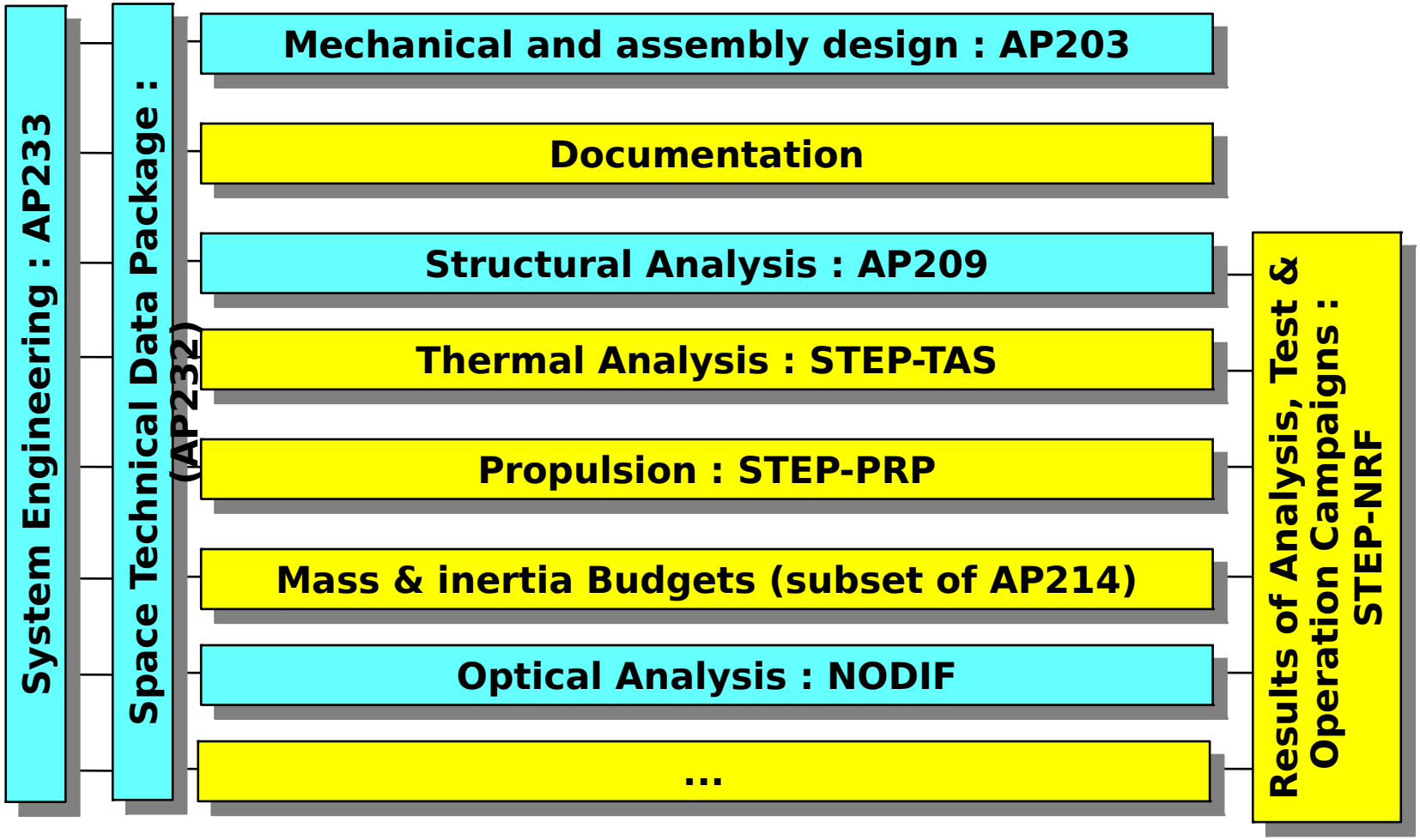
Machining
- **Standard:**: STEP-NC/AP224,
- AP 238, and AP2xx (process plan)
- **Status:: In Development / Prototyped**
- **Software:**: Gibbs, STEP-Tools, CAMsoft
- **Orgs:** Honeywell, Boeing, GM, NASA-JPL

Inspection (Off-line)
- **Standard**: AP219
- **Status:** In Development
- **Software**: Technomatics, Brown, eSharp
- **Orgs: NIST, DMIS, Boeing, Chrysler, AIAG**

Life-Cycle Management
- **Standard**: PLCS / AP 239
- **Status:** In Development
- **Software**: PTC, LSC, AEI, Bonn
- **Orgs:** BAE SYSTEMS, Boeing, Eurostep, NATO, UK MoD

AP233
System Engineering

AP-233 and the STEP architecture

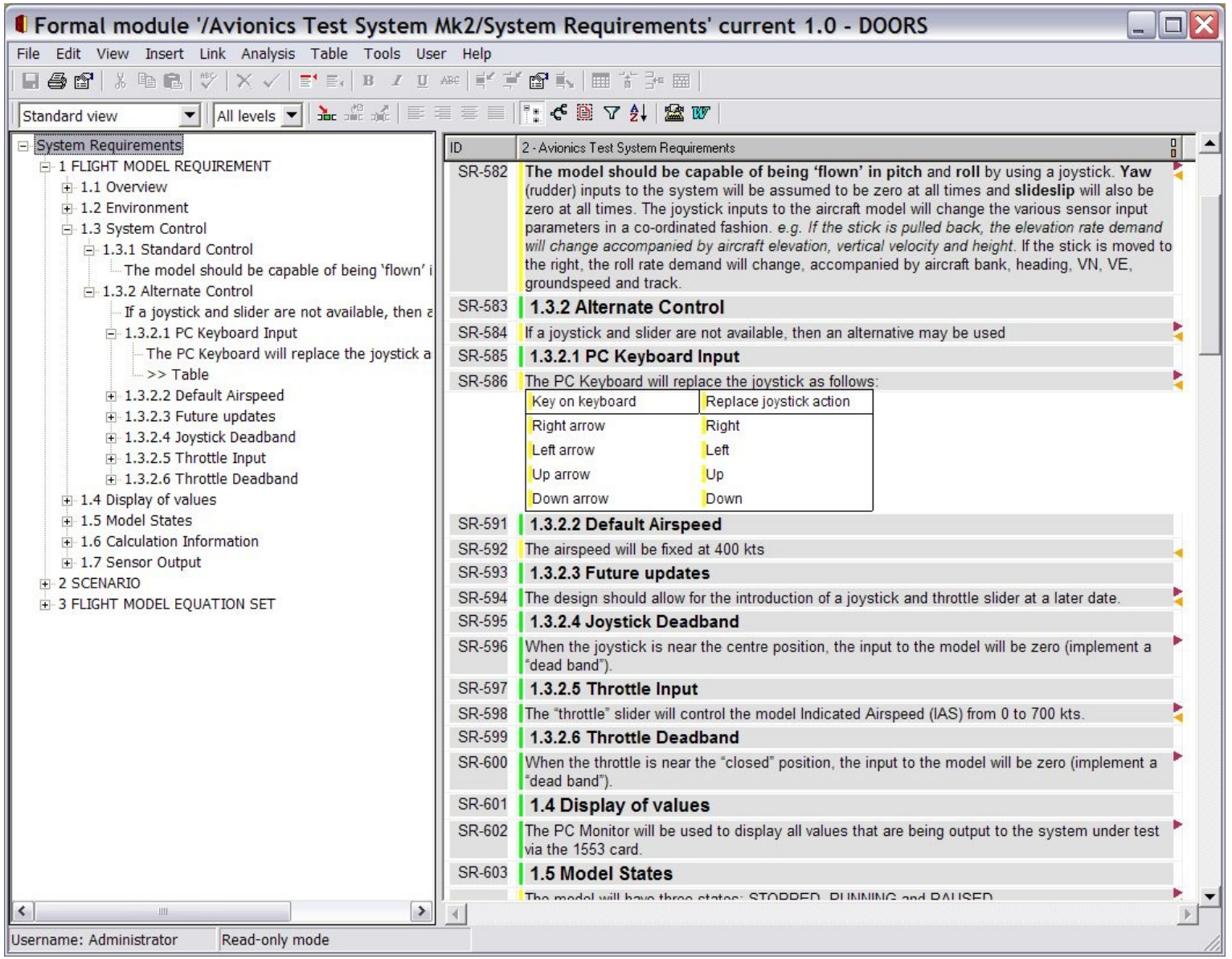

System Engineering : AP233
Space Technical Data Package : (AP232)
Mechanical and assembly design : AP203
Documentation
Structural Analysis : AP209
Thermal Analysis : STEP-TAS
Propulsion : STEP-PRP
Mass & inertia Budgets (subset of AP214)
Optical Analysis : NODIF
...
Results of Analysis, Test & Operation Campaigns : STEP-NRF

Tools used in testing AP233 Requirements Exchanges

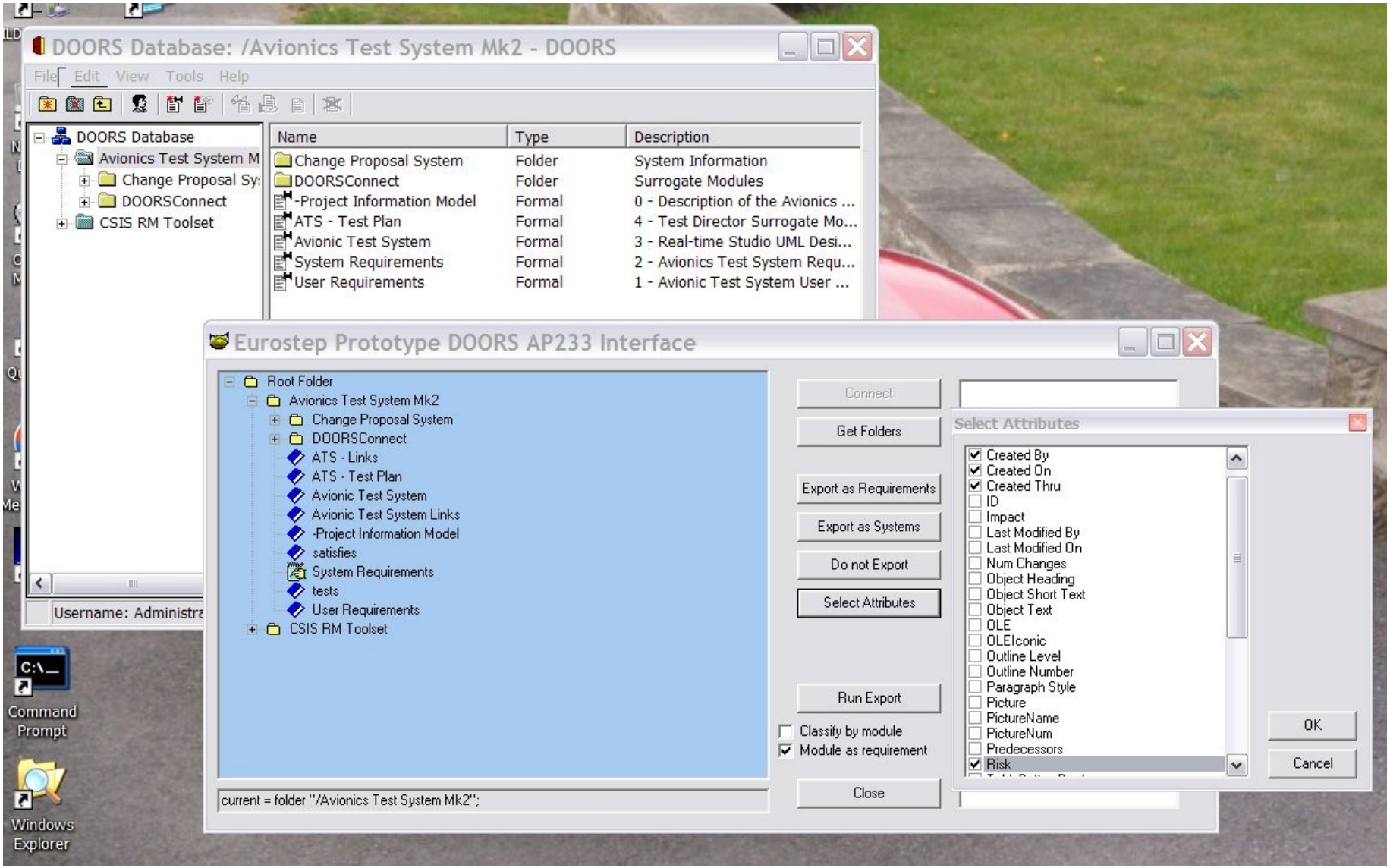
- Eurostep AP233 Demonstrator Tool
- Telelogics DOORS
- Rational RequisitePro
- EDS SLATE
- Vitech CORE
- LKSoft STEP Book
- NIST Expresso
- Eurostep SAS
- MS Word
- MS Excel
- Poseidon UML

AP233 System Engineering

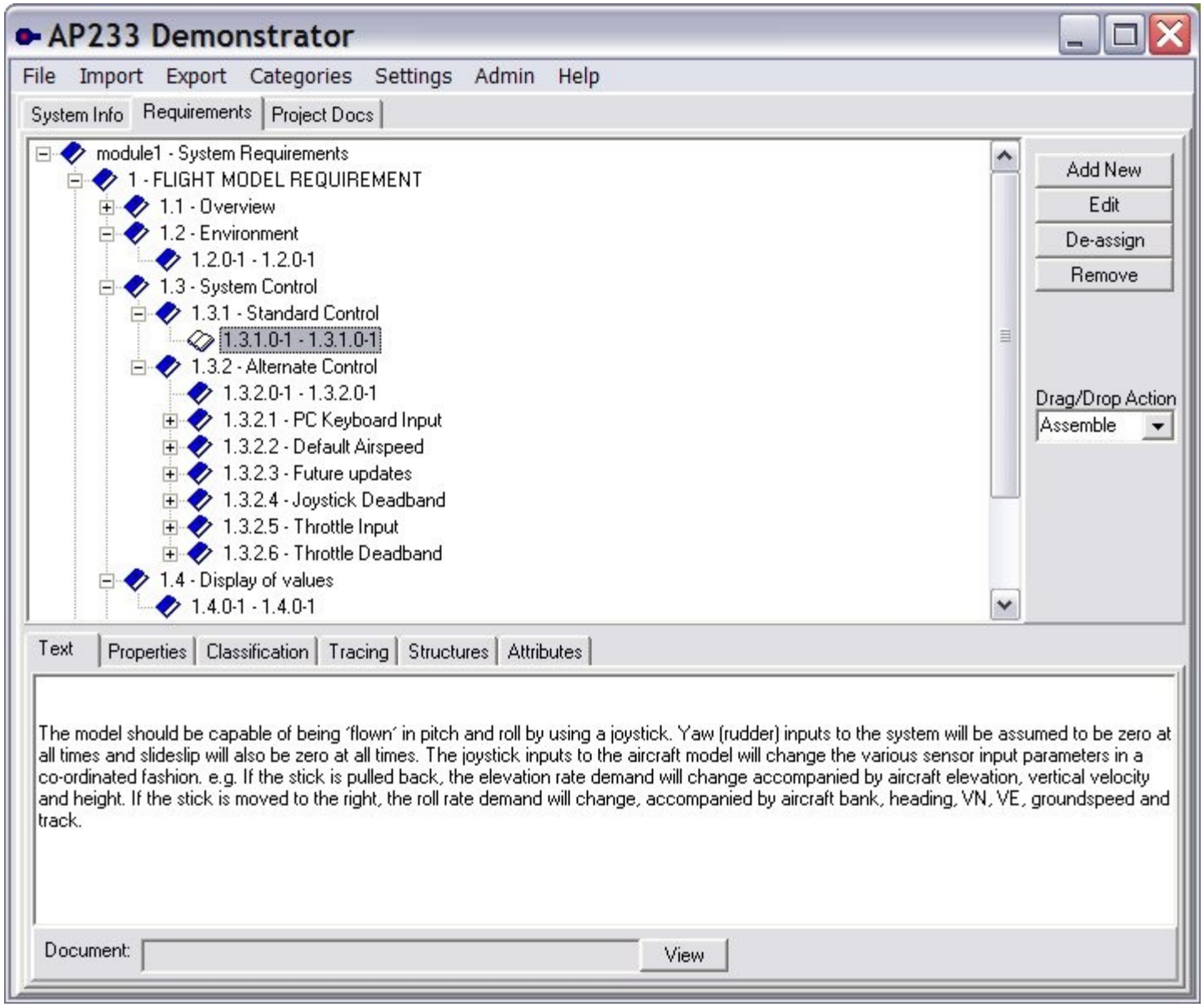
Requirements Initially Developed in DOORS


Formal module '/Avionics Test System Mk2/System Requirements' current 1.0 - DOORS
File Edit View Insert Link Analysis Table Tools User Help
Standard view
All levels
System Requirements
1 FLIGHT MODEL REQUIREMENT
1.1 Overview
1.2 Environment
1.3 System Control
1.3.1 Standard Control
The model should be capable of being 'flown' i
1.3.2 Alternate Control
If a joystick and slider are not available, then a
1.3.2.1 PC Keyboard Input
The PC Keyboard will replace the joystick a
>> Table
1.3.2.2 Default Airspeed
1.3.2.3 Future updates
1.3.2.4 Joystick Deadband
1.3.2.5 Throttle Input
1.3.2.6 Throttle Deadband
1.4 Display of values
1.5 Model States
1.6 Calculation Information
1.7 Sensor Output
2 SCENARIO
3 FLIGHT MODEL EQUATION SET
ID
2 - Avionics Test System Requirements
SR-582
The model should be capable of being 'flown' in pitch and roll by using a joystick. Yaw (rudder) inputs to the system will be assumed to be zero at all times and slideslip will also be zero at all times. The joystick inputs to the aircraft model will change the various sensor input parameters in a co-ordinated fashion. e.g. If the stick is pulled back, the elevation rate demand will change accompanied by aircraft elevation, vertical velocity and height. If the stick is moved to the right, the roll rate demand will change, accompanied by aircraft bank, heading, VN, VE, groundspeed and track.
SR-583
1.3.2 Alternate Control
SR-584
If a joystick and slider are not available, then an alternative may be used
SR-585
1.3.2.1 PC Keyboard Input
SR-586
The PC Keyboard will replace the joystick as follows:
Key on keyboard
Replace joystick action
Right arrow
Right
Left arrow
Left
Up arrow
Up
Down arrow
Down
SR-591
1.3.2.2 Default Airspeed
SR-592
The airspeed will be fixed at 400 kts
SR-593
1.3.2.3 Future updates
SR-594
The design should allow for the introduction of a joystick and throttle slider at a later date.
SR-595
1.3.2.4 Joystick Deadband
SR-596
When the joystick is near the centre position, the input to the model will be zero (implement a "dead band").
SR-597
1.3.2.5 Throttle Input
SR-598
The "throttle" slider will control the model Indicated Airspeed (IAS) from 0 to 700 kts.
SR-599
1.3.2.6 Throttle Deadband
SR-600
When the throttle is near the "closed" position, the input to the model will be zero (implement a "dead band").
SR-601
1.4 Display of values
SR-602
The PC Monitor will be used to display all values that are being output to the system under test via the 1553 card.
SR-603
1.5 Model States
Username: Administrator
Read-only mode

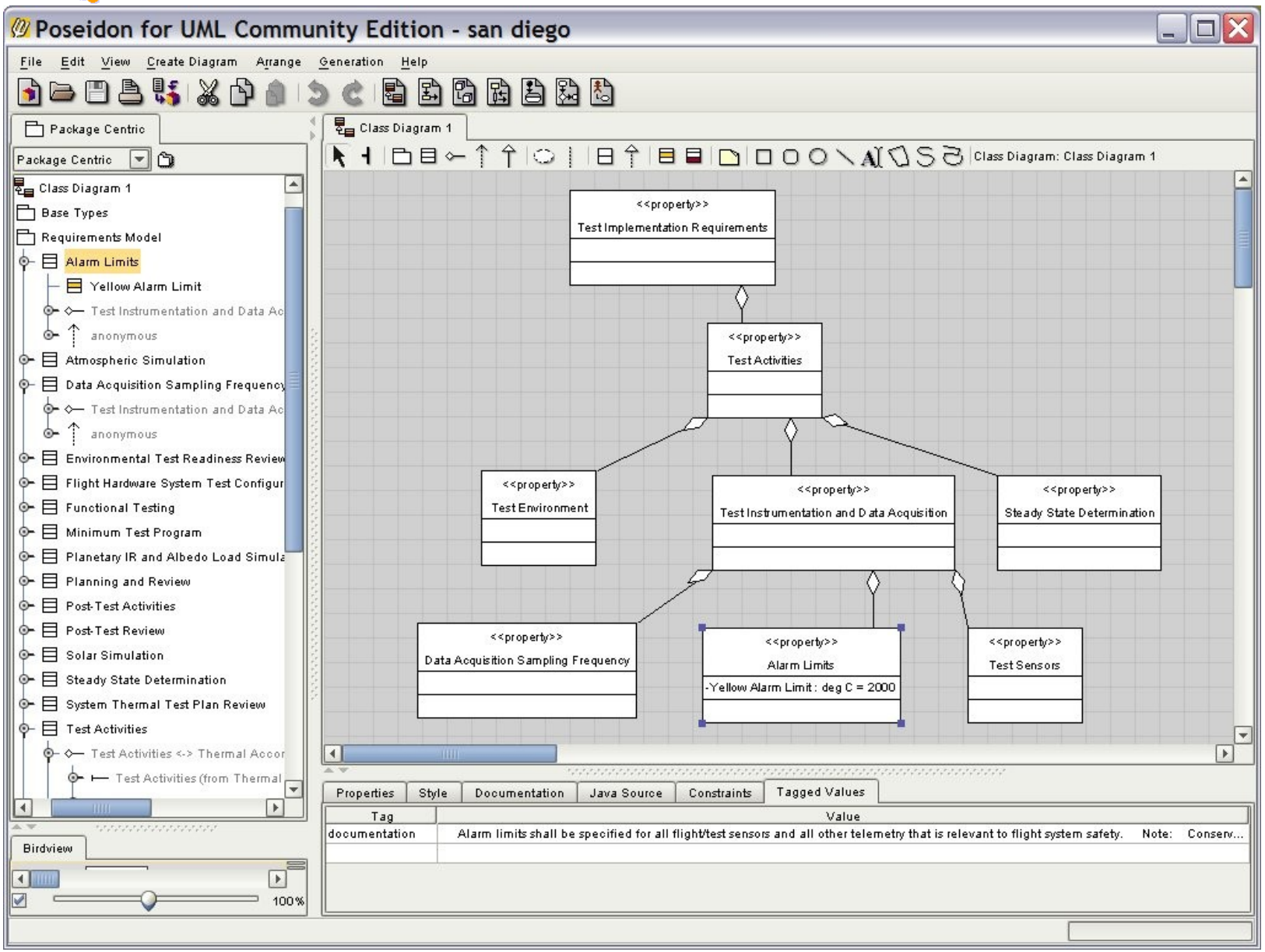
Export from DOORS to AP233 Requirements Format



Requirements in AP233 Demonstrator Tool

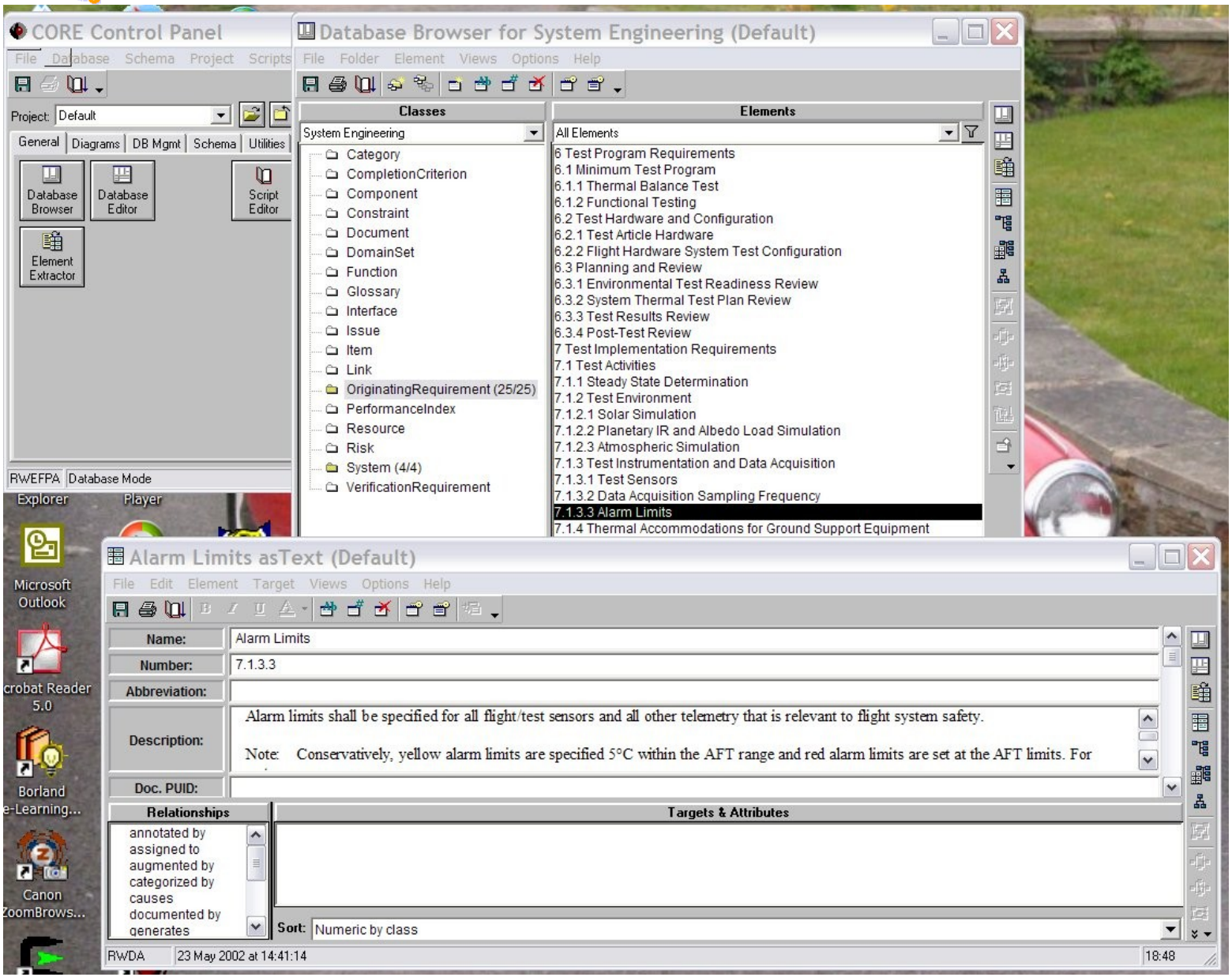

AP233 Demonstrator
File
Import
Export
Categories
Settings
Admin
Help
System Info
Requirements
Project Docs
module1 - System Requirements
1 - FLIGHT MODEL REQUIREMENT
1.1 - Overview
1.2 - Environment
1.2.0-1 - 1.2.0-1
1.3 - System Control
1.3.1 - Standard Control
1.3.1.0-1 - 1.3.1.0-1
1.3.2 - Alternate Control
1.3.2.0-1 - 1.3.2.0-1
1.3.2.1 - PC Keyboard Input
1.3.2.2 - Default Airspeed
1.3.2.3 - Future updates
1.3.2.4 - Joystick Deadband
1.3.2.5 - Throttle Input
1.3.2.6 - Throttle Deadband
1.4 - Display of values
1.4.0-1 - 1.4.0-1
Add New
Edit
De-assign
Remove
Drag/Drop Action
Assemble
Text
Properties
Classification
Tracing
Structures
Attributes
The model should be capable of being 'flown' in pitch and roll by using a joystick. Yaw (rudder) inputs to the system will be assumed to be zero at all times and slideslip will also be zero at all times. The joystick inputs to the aircraft model will change the various sensor input parameters in a co-ordinated fashion. e.g. If the stick is pulled back, the elevation rate demand will change accompanied by aircraft elevation, vertical velocity and height. If the stick is moved to the right, the roll rate demand will change, accompanied by aircraft bank, heading, VN, VE, groundspeed and track.
Document:
View

Requirements imported in Poseidon UML

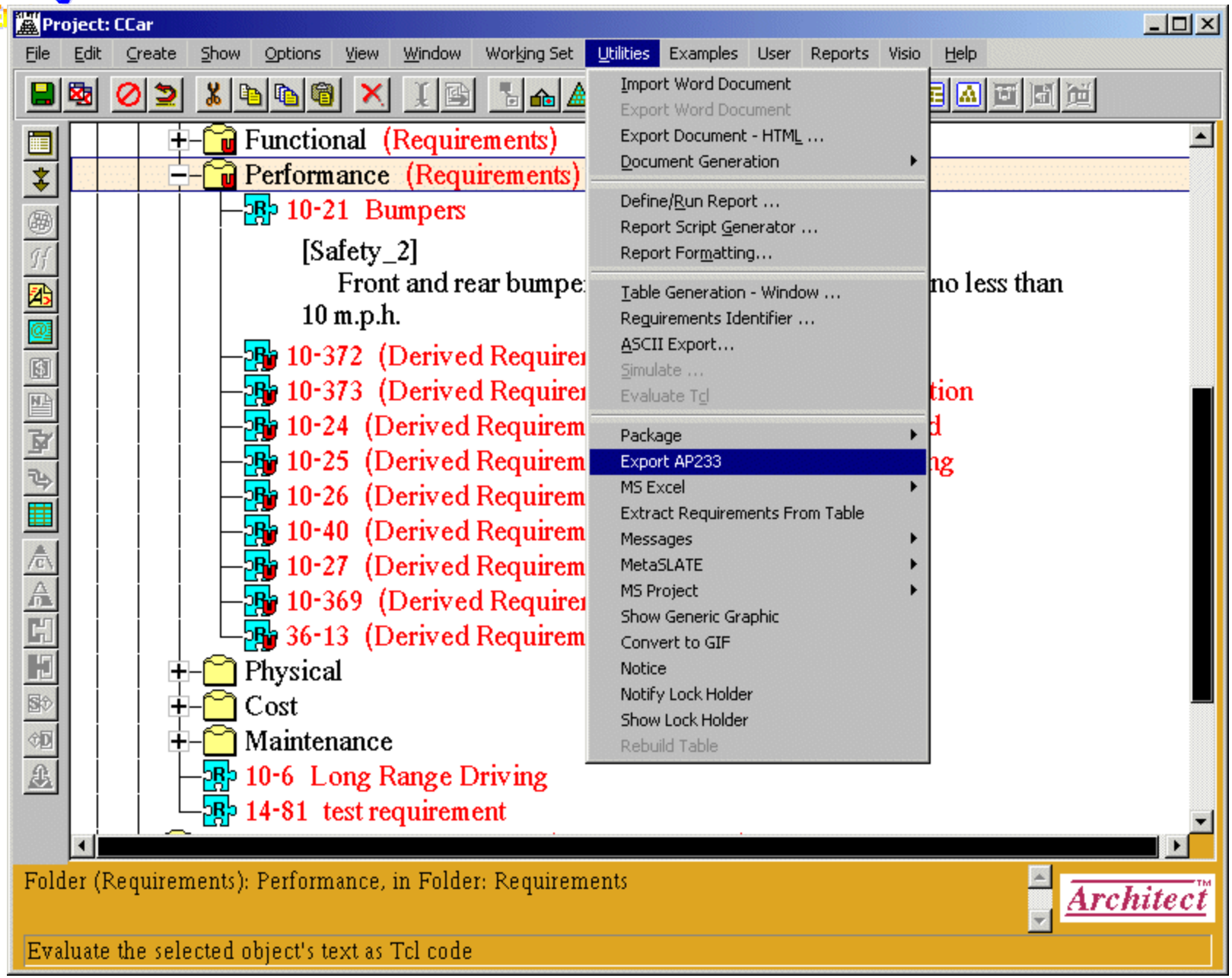


AP233 System Engineering

Requirements imported from AP233 into Vitech CORE



Export from EDS Slate to AP233 Requirements Format



AP233 Requirements Part 21 file (ARM based)

Generated by Eurostep Demonstrator Tool

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('Ian Bailey'), '2;1');
FILE_NAME ('C:\\Program Files\\Eurostep\\AP233 Demonstrator\\Elevator.stp', '2003-05-07T14:02:48', (''), ('Ian Bailey'),
  'ATBX V1.0 - AP233_PBR_ALPHA1 Toolbox Version 1.0 (2002-07-02)', 'ATBX V1.0', 'C:\\Program Files\\Eurostep\\AP233
Demonstrator\\Elevator.stp.log');
FILE_SCHEMA (('AP233_PBR_ALPHA1'));
ENDSEC;
DATA;
#1 = PRODUCT_CATEGORY ('requirement', 'requirement', 'a required property or functionality');
#2 = PRODUCT_CATEGORY ('system', 'system', 'An assembly of interacting, active components or elements forming a whole');
#3 = VIEW_DEFINITION_CONTEXT ('Systems Engineering View', $, 'System Design Stage');
#4 = REPRESENTATION_CONTEXT ('SysEng', 'Systems Engineering');
#8 = PRODUCT_CATEGORY_ASSIGNMENT (#2, (#5));
#12 = PRODUCT_CATEGORY_ASSIGNMENT (#1, (#9));
#36 = PRODUCT_CATEGORY_ASSIGNMENT (#1, (#33));
#46 = PRODUCT_CATEGORY_ASSIGNMENT (#1, (#43));
#56 = PRODUCT_CATEGORY_ASSIGNMENT (#1, (#53));
#70 = PRODUCT_CATEGORY_ASSIGNMENT (#1, (#67));
#79 = PRODUCT_CATEGORY_ASSIGNMENT (#1, (#76));
#90 = PRODUCT_CATEGORY_ASSIGNMENT (#1, (#87));
#135 = SYSTEM_DESIGN ((#3), #134, '', 'MP1v1.0view1', #3, 'MP1v1.0view1', .F.);
#134 = SYSTEM_VERSION ('1.0', 'version 1.0 of MP1', #133);
#133 = SYSTEM ('Description for Maintenance panel', 'MP1', 'Maintenance panel');
#140 = SYSTEM_DESIGN ((#3), #139, '', 'DT1v1.0view1', #3, 'DT1v1.0view1', .F.);
#139 = SYSTEM_VERSION ('1.0', 'version 1.0 of DT1', #138);
#138 = SYSTEM ('Description for Diagnostic tool', 'DT1', 'Diagnostic tool');
#161 = PRODUCT_CATEGORY_ASSIGNMENT (#2, (#158));
#142 = SYSTEM_ASSEMBLY_RELATIONSHIP ($, 'Diagnostic tool', #140, #125, 'System Assembly', $, $, $);
#137 = SYSTEM_ASSEMBLY_RELATIONSHIP ($, 'Maintenance panel', #135, #125, 'System Assembly', $, $, $);
#9 = REQUIREMENT ('High reliability', 'R1', 'Reliabiity');
#33 = REQUIREMENT ('Lift 200 people', 'R2', 'Lift capability');
#43 = REQUIREMENT ('Maximum number of passengers', 'R3', 'Maximum passenger load');
#53 = REQUIREMENT ('Maximum weight', 'R4', 'Maximum weight');
#67 = REQUIREMENT ('Speed requirements', 'R5', 'Speed');
#5 = SYSTEM ('A state-of-the-art elevator', 'EV1', 'Elevator');
#6 = SYSTEM_VERSION ('1.0', 'version 1.0 of EV1', #5);
#7 = SYSTEM_DESIGN ((#3), #6, '', 'EV1v1.0view1', #3, 'EV1v1.0view1', .F.);
```

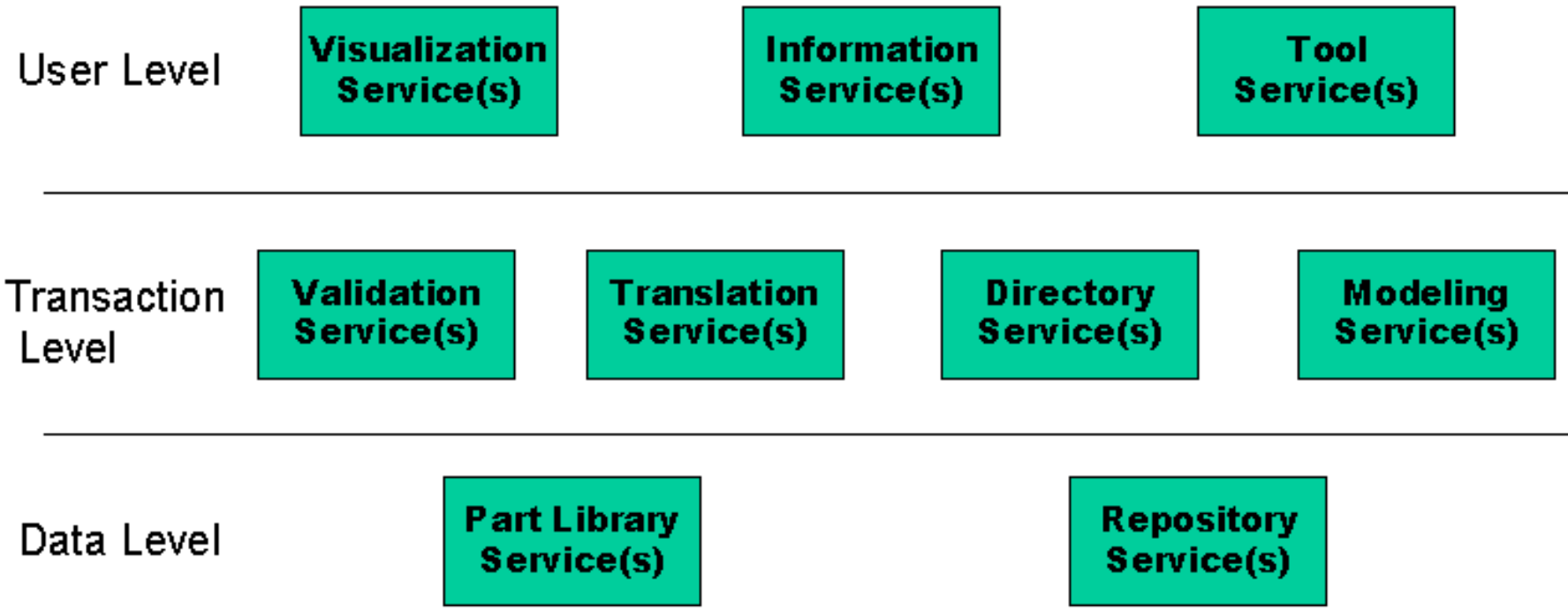
Where do we go from here?

- continue to pursue sponsorship
  - with US agencies and organization
  - with European organizations
- continue collaboration between OMG SE DSIG, INCOSE MDSD and STEP AP233 teams
- extend areas of collaborative wo through the PDES Inc consortium

AP233
System Engineering

[Backup Slides]

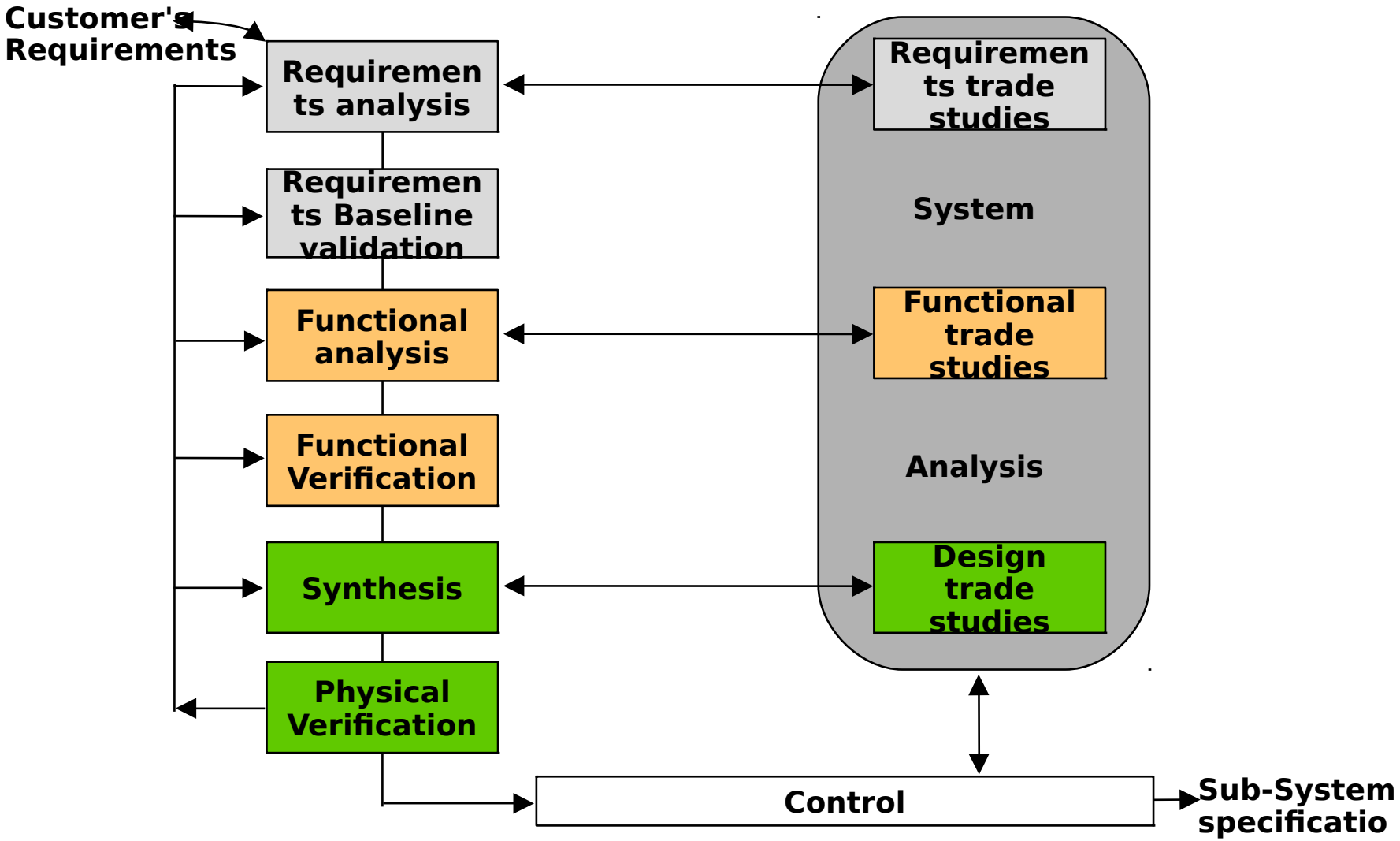
High-level Diagram of Infrastructure Services
to support
Information Access, Reuse and Integration



Infrastructure Services should have:
1) standard interfaces
2) distributable features
3) platform independance

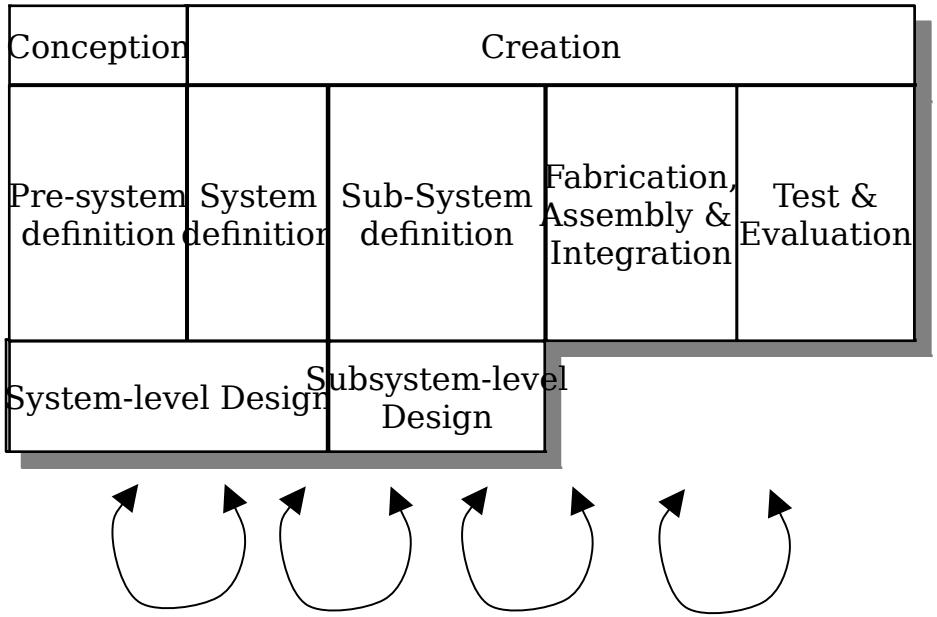
AP233
System Engineering

SYSTEM ENGINEERING PROCESS
(this example from IEEE1220)

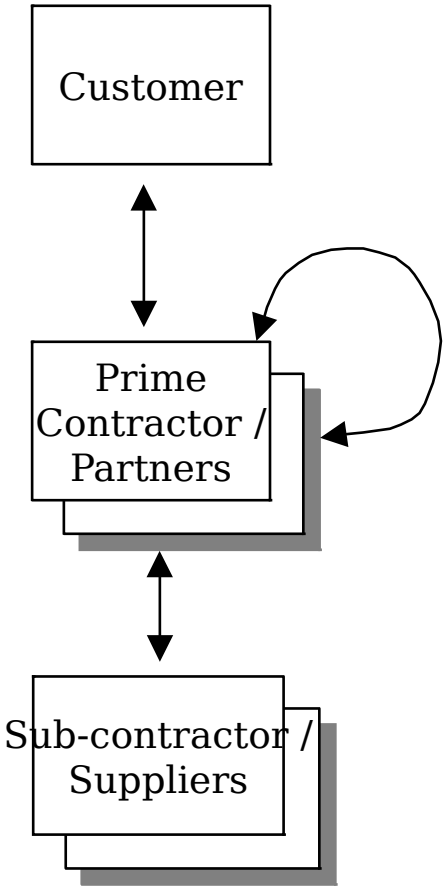


The situation…
Data Exchanges in context

Development life-cycle phases

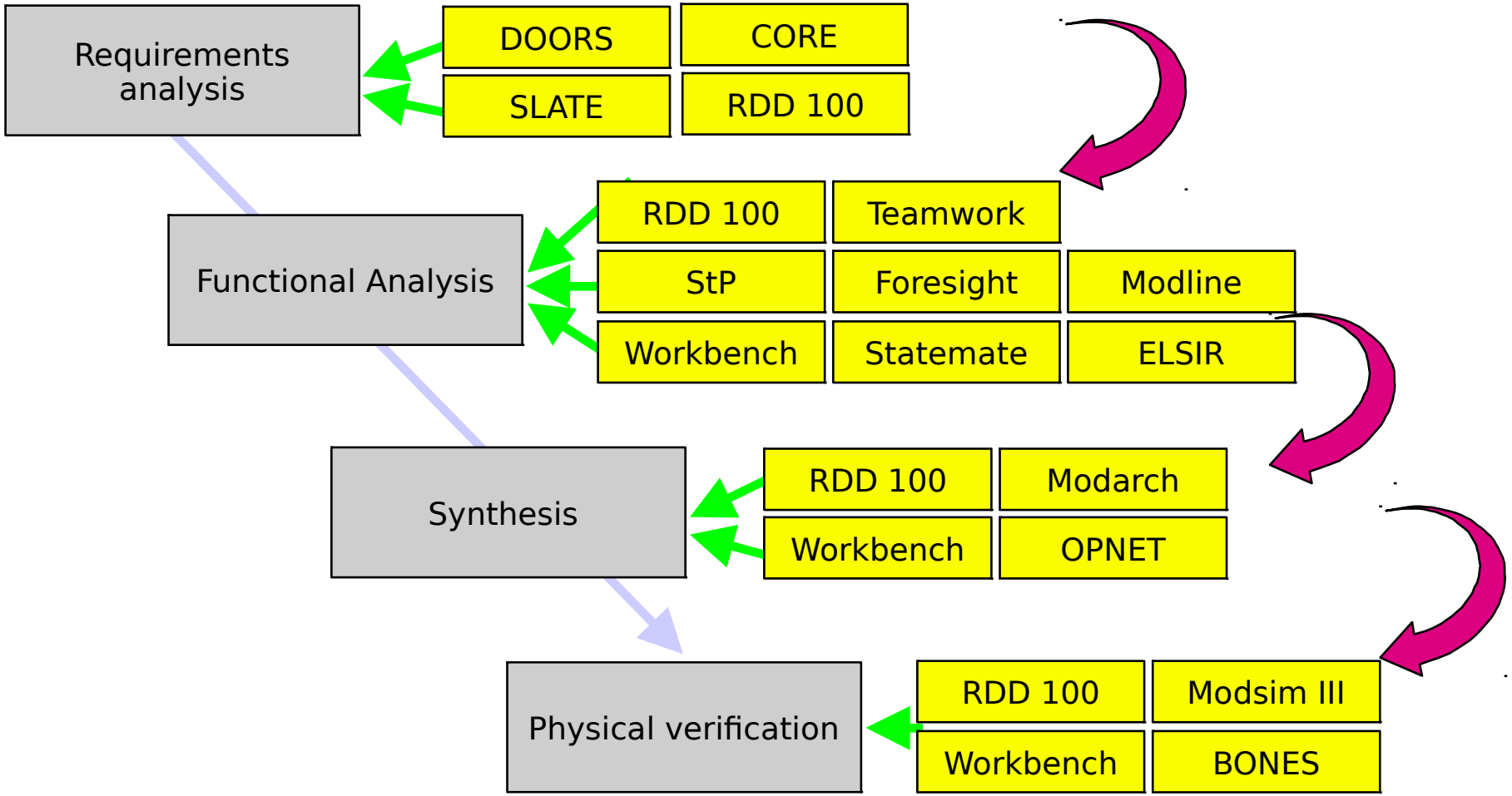


Fundamental Data exchanges



AP233
System Engineering

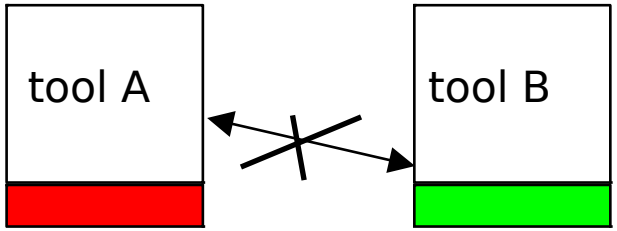
SE Process and SE Tools



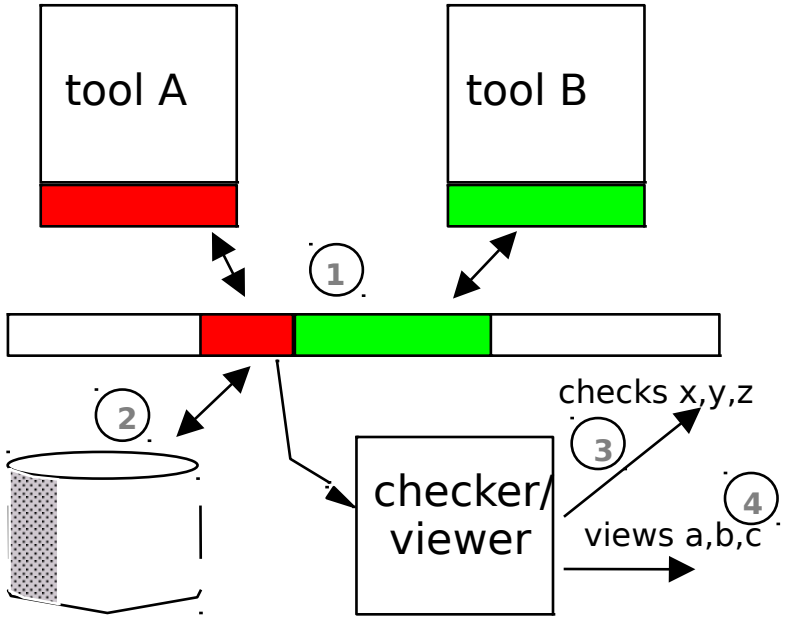
AP233
System Engineering

From isolation to inter-working

The Past



The Future



The consequences:
Reduced benefits from each tool;
Costs migrating to new tools/versions;
Lack of an 'integrated' design dataset;
Compromised success of partnerships;

The benefits:
Reduced costs to transfer & check data (1, 3);
Better achieve a coherent design (3, 4);
Facilitate Integrated Product Development (1, 3, 4)
Facilitate documentation/ design data managemen

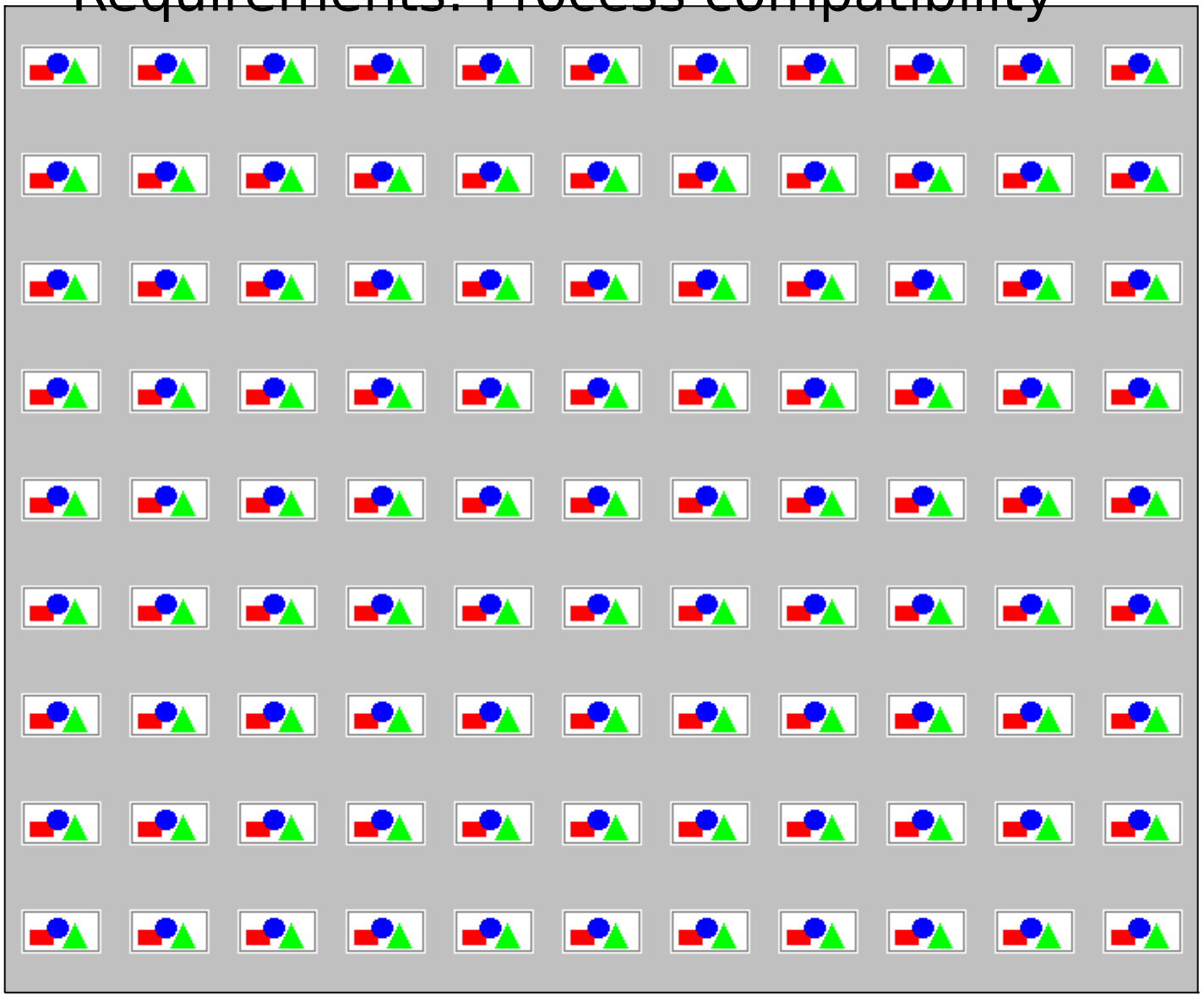
AP233
System Engineering

Requirements for a system engineering exchange standard

- **Shall be compliant with SE processes**
  - > **Support for primary, support & organisational processes**
  - > **From requirement elicitation to V&V**
  - > **Consistent with concepts of SEMP**
  - > **Through life cycle support**
- **Shall provide a tool independent representation**
- **Shall be compatible with industrial organisation**
- **Shall be implementable / adoptable by vendors**
- **Lead to a consequential reduction in number / variety of design tool interfaces**

AP233
System Engineering

Requirements: Process compatibility



AP233
System Engineering

Requirements: Process - view points

- **Requirement point of view**
- **Functional structure point of view**
- **Physical structure & allocation point of view**
- **Configuration and traceability point of view**
- **Project & data management point of view**

AP233
System Engineering

Requirements: Process - Requirements

- **Requirements, categories and structure**
- **System context & operational modes**
- **System environment description**
- **Validation & verification information**
- **Implementation verification**
- **Requirement trade study information**
- **Links to:**
  - **maturity, design phases, project control, project organisation, documentation support**

AP233
System Engineering

Requirements: Process - Functionality

- **Functional elements and child-parent hierarchy**
- **Requirement allocation**
- **Function inputs and function outputs**
- **Data description**
- **Behaviour description**
  - **possible modelling paradigms: state machines, time lines, structured text, logic tables, mathematical, analytical..**
- **Links to:**
  - **maturity, design phases, project control, project organisation, documentation support**

AP233
System Engineering

Requirements: Process - Physical structure

- **Component description**
- **Subsystem hierarchy definition**
- **Data links / Physical interface definition**
- **Information interface definition**
- **Performance allocation**
- **Function allocation**
- **Support for validation, verification, traceability**
- **Links to:**
  - **maturity, design phases, project control, project organisation, documentation support**

AP233
System Engineering

Requirements: Process - Configuration and traceability

- **Item identification and version control**
- **Analysis iteration control with link to version**
- **Traceability management (upward / backward)**
- **Justification traceability**
- **Security management**
- **Link to full product management (consistency between real product and architectures)**
- **Trade-off & Justification support**
- **Exchange control**

AP233
System Engineering

Requirements: Process - Project & data management

- **Design process**
  - **>Support for work breakdown structure**
  - **>Support for a flexible process representation**
- **Partner organisation, stakeholders**
- **Design product packaging**
- **Work allocation**

AP233
System Engineering

Requirements: Tool independence

- **Shall provide a tool independent representation**

==>

- **>Neutral format data exchanges**
- **>Neutral modelling paradigms**
- **>Flexible item representation and description**
- **>Extendibility**
- **>Long term design-data storage**
- **>Compatibility with several technology platforms**
  - **Upward compatibility with new enabling technologies (distributed environments, distributed repositories...)**
  - **Backward compatibility with simple exchange techniques**

AP233
System Engineering

Requirements: Organisation

- **Shall be compatible with industrial organisations**

==>

- **Compatibility with industrial adopted technologies for data sharing & exchange**
- **Support for organisation description and work allocation**
- **User oriented / Usable**
- **Supports both high / low data volumes; ‘deltas’**
- **Supports data exchange management**
- **Compatibility with other techniques used in industrial domains (CAD systems…)**
- **Extensible - evolving processes / data types**

AP233
System Engineering

Requirements: Vendor acceptance

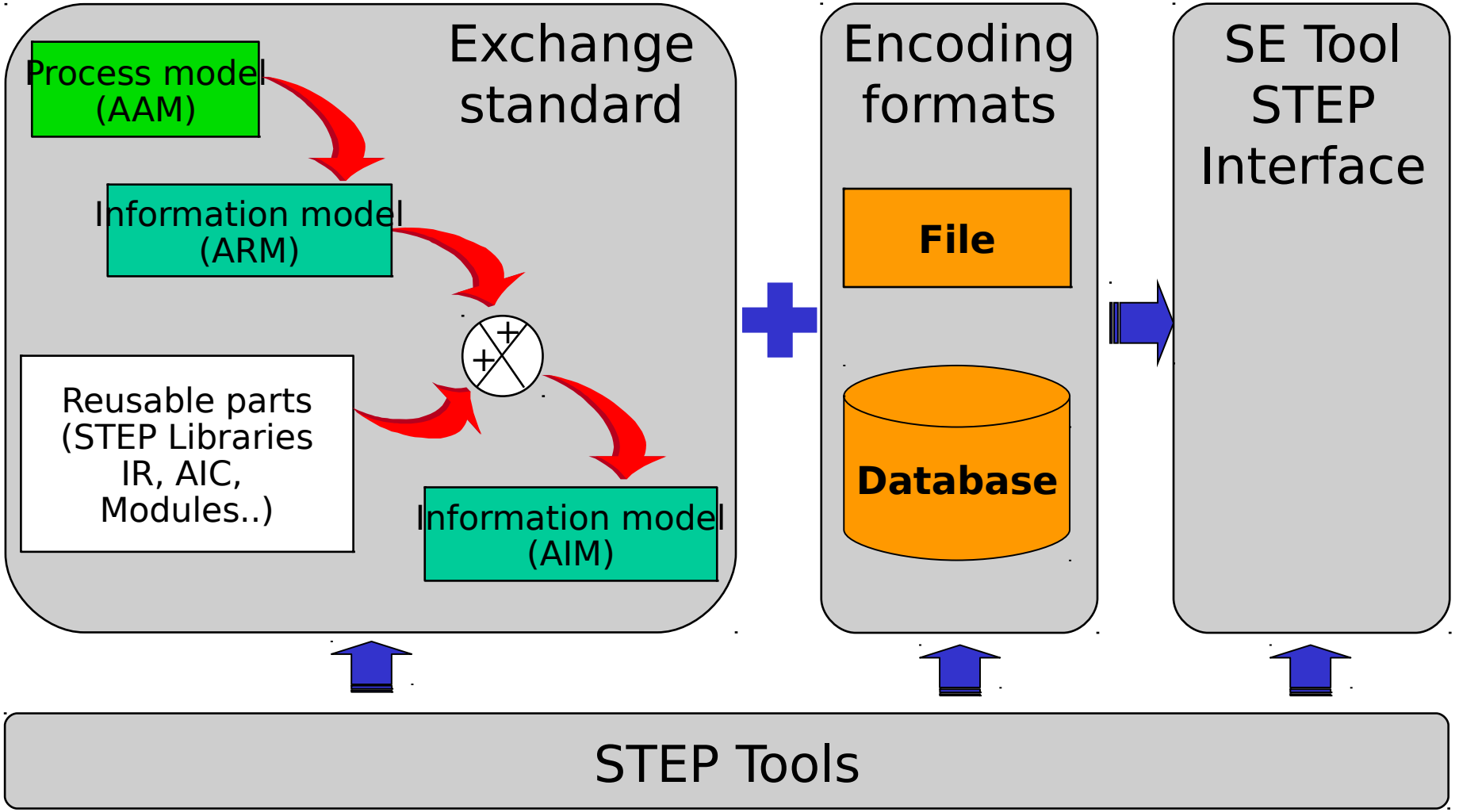
- **Shall be implementable / adoptable by vendors**

==>

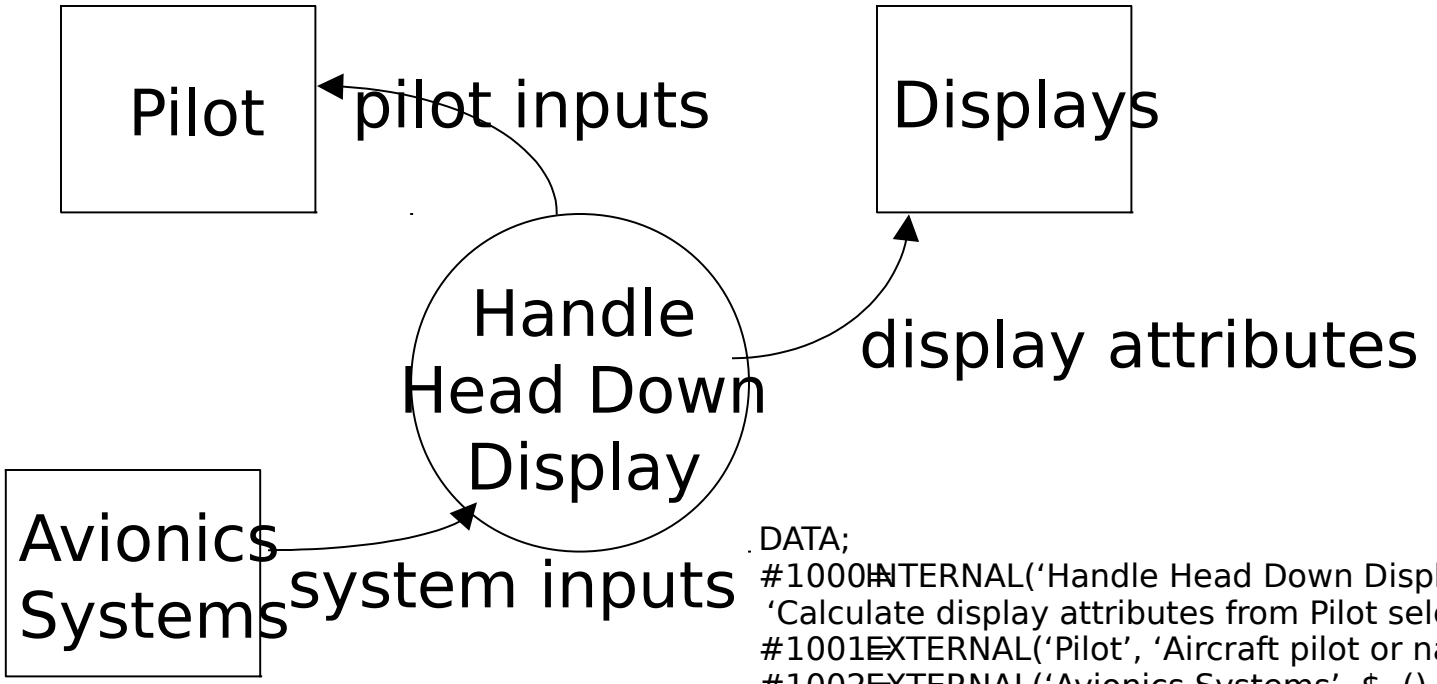
- **Shown to be implementable**
- **Feasible to support (effort / cost / ROI)**
- **Tool neutral / vendor independent**
- **Based on mature data exchange technology**
- **Widely supported by tools & consultancy**
- **Widely supported by tool market users**

AP233
System Engineering

Interface development process



Example design encoding (Part 21)




```
DATA;
#1000=INTERNAL('Handle Head Down Display', $, (#1004, #1005), (#
 'Calculate display attributes from Pilot selections and system inputs');
#1001=EXTERNAL('Pilot', 'Aircraft pilot or navigator', (), (#1004));
#1002=EXTERNAL('Avionics Systems', $, (), (#1005));
#1003=EXTERNAL('Displays', 'Cockpit displays', (#1006), ());
#1004=FLOW('pilot inputs');
#1005=FLOW('system inputs');
#1006=FLOW('display attributes');
END_DATA;
```

AP233
System Engineering

AP233 Basic Philosophy & design principles

- **Focussed on semantic level information**
  - **Graphics Unit of Functionality (UoF) is the exception**
- **Aspects of data model driven from principles**
  - **Distinction between Instances and Definition**
  - **Concept to support ‘re-use’**
- **Aspects driven by ease of model evolution**
  - **Relationship entities**
  - **Support of templates for textual descriptions**
- **Model not tool-specific**

AP233
System Engineering

Requirement UoF (1)

- **Related to Requirement elicitation phase**
- **Defines several kind of requirements**
  - **Functional requirements**
  - **Constraints**
  - **Physical requirements**
  - **Operational requirements**
- **Other categories can be added from**
  - **ECSS 10A**
  - **EIA632…**

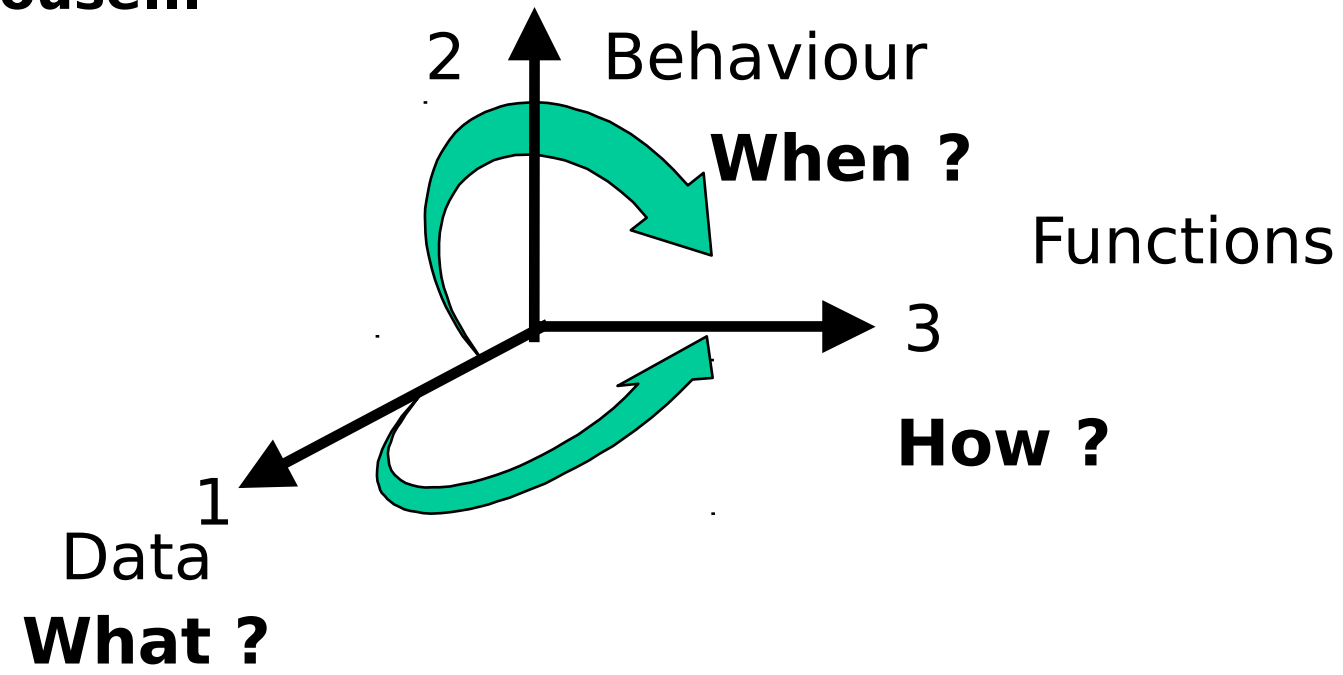
AP233
System Engineering

Requirement UoF (2)

- **Supports operational scenario definition**
- **Supports Derived requirements & composition relationship**
- **Defines the link between external entities and the system to be engineered**
  - **>Externals / functional context**
- **Concept of Traceability matrices support**
  - **>To functional breakdown structure**
  - **>To physical breakdown structure**

Functional UoF (1)

- **Not linked to a particular engineering methodology**
  - > **SART style**
  - > **RDD style**
  - > **In-house…**



Functional UoF (2)

- **Supports the functional breakdown structure**
  - **Functions, instance & definition**
  - **Hierarchical relationships**
  - **Flows**
    - **Flow groups**
    - **Stores**
- **Data description**
  - **Data structure**
  - **Data types**
- **Links to Behavioural model**
  - **Causal chain, Finite state machine, synchronous**

AP233
System Engineering

Behaviour UoF

- **Finite state machines (State chart style)**
  - **State, instance & definition**
  - **State hierarchies**
  - **Transition**
  - **Action on transition**
  - **Events**
  - **Control of functions**
- **Causal chains for safety critical systems**
- **Synchronous model (Clock driven behaviour)**

AP233
System Engineering

Physical UoF

- **Component definition and breakdown structure**
  - > **Product Breakdown Structure (PBS)**
  - > **Bill of Materials (BOM)**
- **Physical path description**
- **Functional/physical mapping**

  Function <=> Physical component

  Flows <=> Physical links

AP233
System Engineering

Graphics UoF

- **Objective:**
  - **ensure (where applicable) that designs on receiving tool can bear ‘similarity’ in layout to original**
- **Visual Elements**
  - **Nodes, links that appear on SE diagrams**
  - **Association to semantic elements**
  - **Graphics points (nodes, links, link path) for diagram layout**
- **Is this the most appropriate approach?**

AP233
System Engineering

System configuration management UoF (1)

- **Configuration Management Item & Item Id**
- **Traceability matrices support**
  - **From requirements through to physical components**
- **Documentation support (‘Package’)**
- **Support for version control**
  - **release / approval**
  - **versions**
  - **variants**
  - **workspace revision**

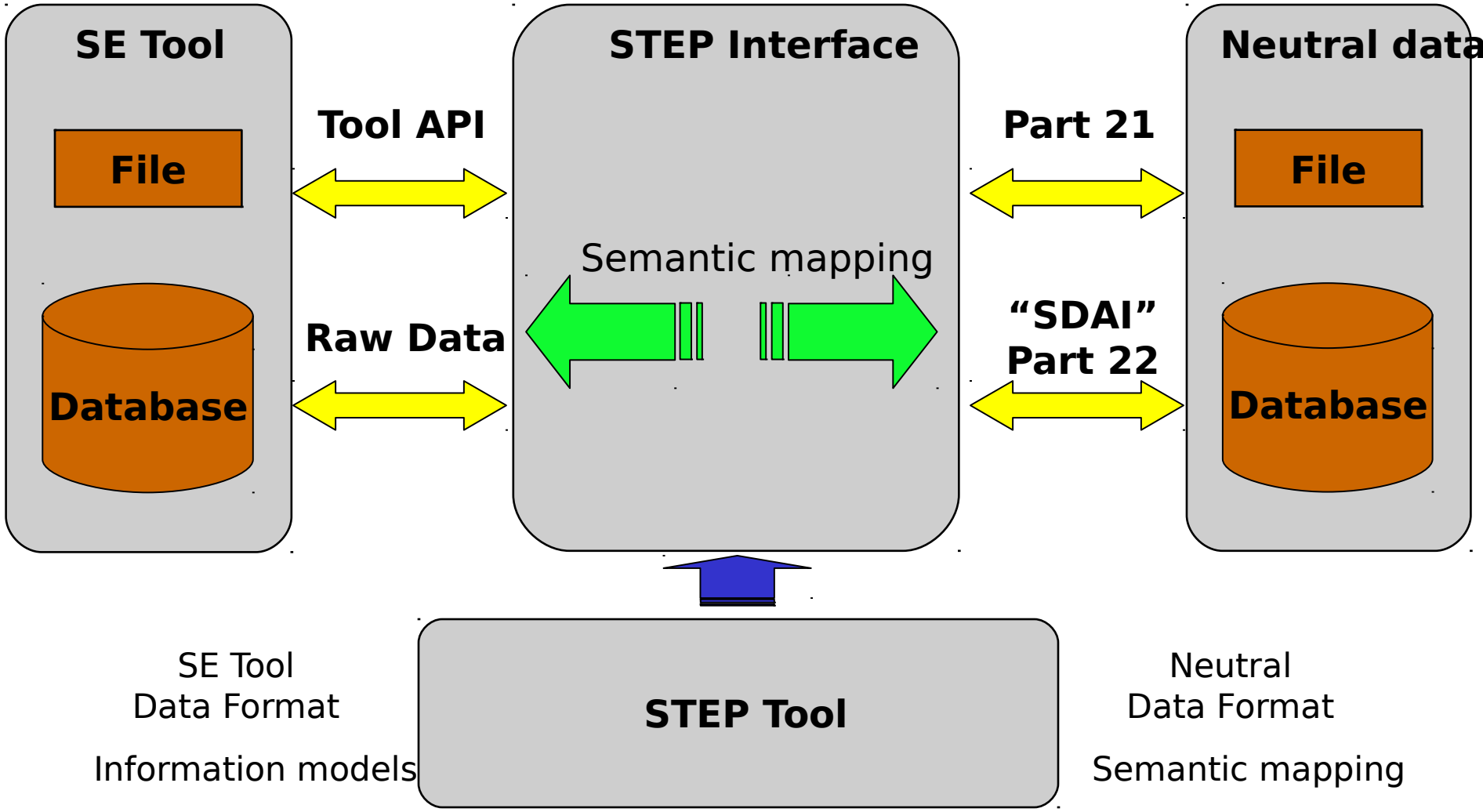
AP233
System Engineering

System configuration management UoF (2)

- **Support for justification and maturity indices**
- **Relationships to process**
  - **work breakdown, activities & products**
- **Support for industrial schema**
  - ***"Who does what and where"***
  - **Project**
  - **Company Id**
  - **Person Id**
- **Convergence with STEP Product Data Mgmt (PDM) Schema**

AP233
System Engineering

STEP Interface usage w.r.t SE Tools



AP233
System Engineering

Test & Evaluation: ‘the bottom line’

- **Actual measurements of limited exchanges capture how time spent**
- **Simple analysis indicates projected times**
- **Note several activities currently due to prototype technology:**
  - **Produce Part21 file..**
  - **Manage & transfer..**
  - **Prepare for import**
  - **..are likely to go to zero in industrial implementation**